What is claimed is:

1. A method comprising:

identifying a representation of a binary image in a pixel matrix, wherein the pixel matrix comprises a plurality of portions;

computing the number of runs for a first portion of the pixel matrix, wherein a run is a maximal sequence of pixels having a predetermined value in the first portion;

computing the number of neighboring runs between the first portion and a second portion of the pixel matrix, wherein a neighboring run is a run in which at least one pixel of the run is in the neighborhood of a run in an adjacent portion; and

computing the Euler number from the number of runs and the number of neighboring runs.

2. The method of claim 1, computing the number of runs for a first portion of the pixel matrix further comprising:

identifying one or more runs of the first portion;

counting the number of runs.

3. The method of claim 2, computing the number of neighboring runs between the first portion and a second portion of the pixel matrix further comprising:

determining a neighborhood size;

identifying a second run in the second portion; and

determining whether at least one pixel of the second run is in the neighborhood of the run.

4. The method of claim 3, computing the Euler number from the number of runs and the number of neighboring runs further comprising:

subtracting the number of neighboring runs between the first portion and the second portion from a sum of the number of runs in the first portion and the second portion to arrive at a result; and

adding the result to an Euler number for a third portion.

5. The method of claim 4, computing the Euler number from the number of runs and the number of neighboring runs further comprising:

subtracting from the result the number of neighboring runs between the third portion and a prior third portion.

6. The method of claim 5, identifying the binary image in a pixel matrix further comprising identifying pixels of the predetermined value.

7. A system comprising:

a run processor to compute a run number, wherein the run number is the number of runs in a portion of a pixel matrix; and

a neighboring run processor to compute a neighboring run number, wherein the neighboring run number is the number of neighboring runs between the portion and a second portion of the pixel matrix and the neighboring run processor receives a plurality of signals from the run processor.

8. The system of claim 7, further comprising:

a second run processor to compute a second run number;

a second neighboring run processor to compute a second neighboring run number; and

a math processor.

9. The system of claim 8, wherein the math processor further comprises:

a run adder to add the run number to the second run number;

a neighboring run adder to add the neighboring run number to the second neighboring run number; and

a subtractor to subtract the run number from the neighboring run number.

10. The system of claim 7, wherein the portions comprise rows of the pixel matrix.

11. The system of claim 7, wherein the portions comprise columns of the pixel matrix.

12. An article comprising a medium storing software for enabling a processor-based system to:

identify a binary image in a pixel matrix, wherein the pixel matrix comprises a plurality of portions;

compute the number of runs for a first portion of the pixel matrix;

compute the number of neighboring runs between the first portion and a second portion of the pixel matrix; and

compute an Euler number from the number of runs and the number of neighboring runs.

13. The article of claim 12, further storing software for enabling a processor-based system to compute the number of runs for a first portion of the pixel matrix by:

identifying one or more runs of the first portion, wherein a run is a maximal sequence of pixels with a predetermined value in the first portion;

counting the number of runs.

14. The article of claim 13, further storing software for enabling a processor-based system to compute the number of neighboring runs between the first portion and a second portion of the pixel matrix by:

determining a neighborhood size;

identifying a second run in the second portion; and

determining whether at least one pixel of the second run is in the neighborhood of the run.

15. The article of claim 14, further storing software for enabling a processor-based system to compute the Euler number from the number of runs and the number of neighboring runs by:

subtracting the number of neighboring runs between the first portion and the second portion from a sum of the number of runs in the first portion and the second portion to arrive at a result; and

adding the result to an Euler number for a third portion.

16. The article of claim 15, further storing software for enabling a processor-based system to compute the Euler number from the number of runs and the number of neighboring runs by:

subtracting from the result the number of neighboring runs between the third portion and a prior third portion.